

Sequence-Level Knowledge Distillation

Yoon Kim

Alexander M. Rush



HarvardNLP

Code: <https://github.com/harvardnlp/seq2seq-attn>

Sequence-to-Sequence

- Machine Translation (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015; Luong et al., 2015)
- Question Answering (Hermann et al., 2015)
- Conversation (Vinyals et al., 2015a; Serban et al., 2016; Li et al., 2016)
- Parsing (Vinyals and Le, 2015)
- Speech (Chorowski et al., 2015; Chan et al., 2015)
- Summarization (Rush et al., 2015)
- Caption Generation (Xu et al., 2015; Vinyals et al., 2015b)
- Video-Generation (Srivastava et al., 2015)
- NER/POS-Tagging (Gillick et al., 2016)

Sequence-to-Sequence

- Machine Translation (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015; Luong et al., 2015)
- Question Answering (Hermann et al., 2015)
- Conversation (Vinyals et al., 2015a; Serban et al., 2016; Li et al., 2016)
- Parsing (Vinyals and Le, 2015)
- Speech (Chorowski et al., 2015; Chan et al., 2015)
- Summarization (Rush et al., 2015)
- Caption Generation (Xu et al., 2015; Vinyals et al., 2015b)
- Video-Generation (Srivastava et al., 2015)
- NER/POS-Tagging (Gillick et al., 2016)



Google unleashes deep learning tech on language with **Neural** ...

[TechCrunch](#) - Sep 27, 2016

Google has been working on a **machine learning translation** technique for years, and today is its official debut. The Google **Neural Machine** ...

Google **Translate** now converts Chinese into English with **neural** ...

[VentureBeat](#) - Sep 27, 2016

Google announces **Neural Machine Translation**

[The Stack](#) - Sep 28, 2016

Google announces **Neural Machine Translation** to improve Google ...

Highly Cited - [ZDNet](#) - Sep 27, 2016

Google is using **Neural Networks** for Chinese to English **machine** ...

Opinion - [Firstpost](#) - Sep 28, 2016

Google announces **neural** network to improve **machine translation**

In-Depth - [Seeking Alpha](#) - Sep 27, 2016



ZDNet



VentureBeat



The Stack



Geektime



Ubergizmo



Science Mag...

[View all](#)

SYSTRAN: 1st software provider to launch a **Neural Machine** ...

[GlobeNewswire \(press release\)](#) - Oct 17, 2016

In December, SYSTRAN will communicate the feedback received on Pure **Neural**™ **Machine Translation**, its roadmap and time to market plan ...

Iconic Integrates Custom **Neural Machine Translation** Into ...

[Slator \(press release\) \(subscription\)](#) - Oct 6, 2016

Dublin – October 6, 2016 – Iconic **Translation Machines (Iconic)**, a leading Irish **machine translation (MT)** software and solutions provider, today ...



Neural Machine Translation

Excellent results on many language pairs, but need large models

- Original seq2seq paper (Sutskever et al., 2014): 4-layers/1000 units
- Deep Residual RNNs (Zhou et al., 2016) : 16-layers/512 units
- Google's NMT system (Wu et al., 2016): 8-layers/1024 units

Beam search + ensemble on top

⇒ Deployment is challenging!

Neural Machine Translation

Excellent results on many language pairs, but need large models

- Original seq2seq paper (Sutskever et al., 2014): 4-layers/1000 units
- Deep Residual RNNs (Zhou et al., 2016) : 16-layers/512 units
- Google's NMT system (Wu et al., 2016): 8-layers/1024 units

Beam search + ensemble on top

⇒ Deployment is challenging!

Related Work: Compressing Deep Models

- **Pruning:** Prune weights based on importance criterion (LeCun et al., 1990; Han et al., 2016; See et al., 2016)
- **Knowledge Distillation:** Train a *student* model to learn from a *teacher* model (Bucila et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015; Kuncoro et al., 2016). (Sometimes called “dark knowledge”)

Other methods:

- low-rank matrix factorization of weight matrices (Denton et al., 2014)
- weight binarization (Lin et al., 2016)
- weight sharing (Chen et al., 2015)

Related Work: Compressing Deep Models

- **Pruning:** Prune weights based on importance criterion (LeCun et al., 1990; Han et al., 2016; See et al., 2016)
- **Knowledge Distillation:** Train a *student* model to learn from a *teacher* model (Bucila et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015; Kuncoro et al., 2016). (Sometimes called “dark knowledge”)

Other methods:

- low-rank matrix factorization of weight matrices (Denton et al., 2014)
- weight binarization (Lin et al., 2016)
- weight sharing (Chen et al., 2015)

Related Work: Compressing Deep Models

- **Pruning:** Prune weights based on importance criterion (LeCun et al., 1990; Han et al., 2016; See et al., 2016)
- **Knowledge Distillation:** Train a *student* model to learn from a *teacher* model (Bucila et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015; Kuncoro et al., 2016). (Sometimes called “dark knowledge”)

Other methods:

- low-rank matrix factorization of weight matrices (Denton et al., 2014)
- weight binarization (Lin et al., 2016)
- weight sharing (Chen et al., 2015)

Related Work: Compressing Deep Models

- **Pruning:** Prune weights based on importance criterion (LeCun et al., 1990; Han et al., 2016; See et al., 2016)
- **Knowledge Distillation:** Train a *student* model to learn from a *teacher* model (Bucila et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015; Kuncoro et al., 2016). (Sometimes called “dark knowledge”)

Other methods:

- low-rank matrix factorization of weight matrices (Denton et al., 2014)
- weight binarization (Lin et al., 2016)
- weight sharing (Chen et al., 2015)

Standard Setup

Minimize NLL

$$\mathcal{L}_{\text{NLL}} = - \sum_t \sum_{k \in \mathcal{V}} \mathbb{1}\{y_t = k\} \log p(w_t = k \mid \mathbf{y}_{1:t-1}, \mathbf{x}; \theta)$$

w_t = random variable for the t -th target token with support \mathcal{V}

y_t = ground truth t -th target token

$\mathbf{y}_{1:t-1}$ = target sentence up to $t - 1$

\mathbf{x} = source sentence

$p(\cdot \mid \mathbf{x}; \theta)$ = model distribution, parameterized with θ

(conditioning on source \mathbf{x} dropped from now on)

Knowledge Distillation (Bucila et al., 2006; Hinton et al., 2015)

- Train a *larger teacher* model first to obtain teacher distribution $q(\cdot)$
- Train a *smaller student* model $p(\cdot)$ to mimic the teacher

Word-Level Knowledge Distillation

Teacher distribution: $q(w_t | \mathbf{y}_{1:t-1}; \theta_T)$

$$\mathcal{L}_{\text{NLL}} = - \sum_t \sum_{k \in \mathcal{V}} \mathbb{1}\{y_t = k\} \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

$$\mathcal{L}_{\text{WORD-KD}} = - \sum_t \sum_{k \in \mathcal{V}} q(w_t = k | \mathbf{y}_{1:t-1}; \theta_T) \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

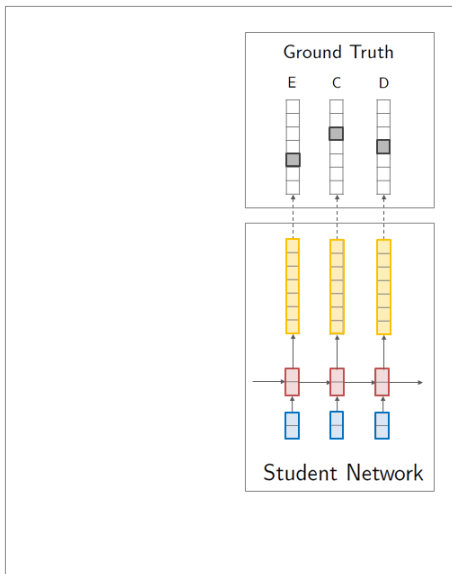
Word-Level Knowledge Distillation

Teacher distribution: $q(w_t | \mathbf{y}_{1:t-1}; \theta_T)$

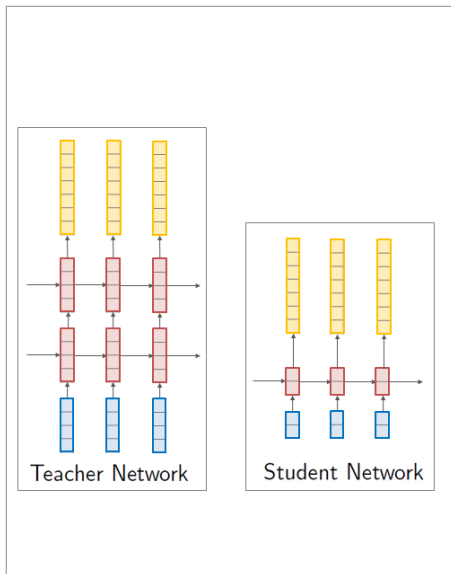
$$\mathcal{L}_{\text{NLL}} = - \sum_t \sum_{k \in \mathcal{V}} \mathbb{1}\{y_t = k\} \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

$$\mathcal{L}_{\text{WORD-KD}} = - \sum_t \sum_{k \in \mathcal{V}} q(w_t = k | \mathbf{y}_{1:t-1}; \theta_T) \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

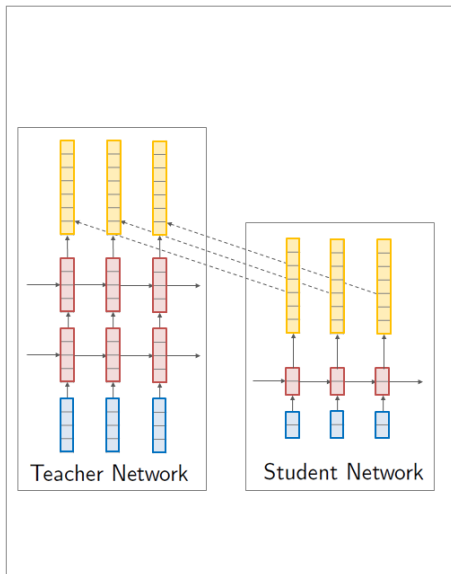
No Knowledge Distillation



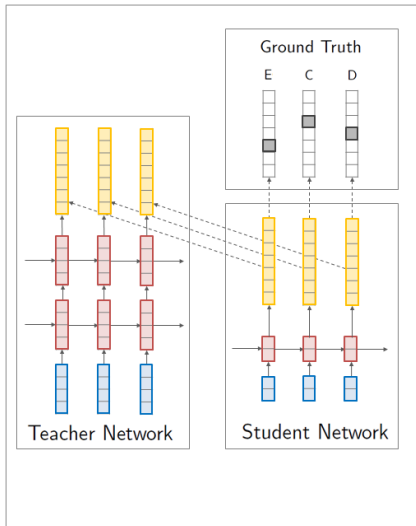
Word-Level Knowledge Distillation



Word-Level Knowledge Distillation



Word-Level Knowledge Distillation



$$\mathcal{L} = \alpha \mathcal{L}_{\text{WORD-KD}} + (1 - \alpha) \mathcal{L}_{\text{NLL}}$$

Word-Level Knowledge Distillation Results

English \rightarrow German (WMT 2014)

Model	BLEU
4×1000 Teacher	19.5
2×500 Baseline (No-KD)	17.6
2×500 Student (Word-KD)	17.7
2×300 Baseline (No-KD)	16.9
2×300 Student (Word-KD)	17.6

This Work

Generalize single-class knowledge distillation to the sequence-level.

- **Sequence-Level Knowledge Distillation (Seq-KD)**: Train towards the teacher's sequence-level distribution.
- **Sequence-Level Interpolation (Seq-Inter)**: Train on a mixture of the teacher's distribution and the data.

Sequence-Level Knowledge Distillation

Recall word-level knowledge distillation:

$$\mathcal{L}_{\text{NLL}} = - \sum_t \sum_{k \in \mathcal{V}} \mathbb{1}\{y_t = k\} \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

$$\mathcal{L}_{\text{WORD-KD}} = - \sum_t \sum_{k \in \mathcal{V}} q(w_t = k | \mathbf{y}_{1:t-1}; \theta_T) \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

Instead of word-level cross-entropy, minimize cross-entropy between q and p implied *sequence*-distributions

$$\mathcal{L}_{\text{NLL}} = - \sum_{\mathbf{w} \in \mathcal{T}} \mathbb{1}\{\mathbf{w} = \mathbf{y}\} \log p(\mathbf{w} | \mathbf{x}; \theta)$$

$$\mathcal{L}_{\text{SEQ-KD}} = - \sum_{\mathbf{w} \in \mathcal{T}} q(\mathbf{w} | \mathbf{x}; \theta_T) \log p(\mathbf{w} | \mathbf{x}; \theta)$$

Sum over an exponentially-sized set \mathcal{T} .

Sequence-Level Knowledge Distillation

Recall word-level knowledge distillation:

$$\mathcal{L}_{\text{NLL}} = - \sum_t \sum_{k \in \mathcal{V}} \mathbb{1}\{y_t = k\} \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

$$\mathcal{L}_{\text{WORD-KD}} = - \sum_t \sum_{k \in \mathcal{V}} q(w_t = k | \mathbf{y}_{1:t-1}; \theta_T) \log p(w_t = k | \mathbf{y}_{1:t-1}; \theta)$$

Instead of word-level cross-entropy, minimize cross-entropy between q and p implied *sequence*-distributions

$$\mathcal{L}_{\text{NLL}} = - \sum_{\mathbf{w} \in \mathcal{T}} \mathbb{1}\{\mathbf{w} = \mathbf{y}\} \log p(\mathbf{w} | \mathbf{x}; \theta)$$

$$\mathcal{L}_{\text{SEQ-KD}} = - \sum_{\mathbf{w} \in \mathcal{T}} q(\mathbf{w} | \mathbf{x}; \theta_T) \log p(\mathbf{w} | \mathbf{x}; \theta)$$

Sum over an exponentially-sized set \mathcal{T} .

Sequence-Level Knowledge Distillation

Approximate $q(\mathbf{w} | \mathbf{x})$ with mode

$$q(\mathbf{w} | \mathbf{x}) \approx \mathbb{1}\{\arg \max_{\mathbf{w}} q(\mathbf{w} | \mathbf{x})\}$$

Approximate mode with beam search

$$\hat{\mathbf{y}} \approx \arg \max_{\mathbf{w}} q(\mathbf{w} | \mathbf{x})$$

Simple model: train the student model on $\hat{\mathbf{y}}$ with NLL

Sequence-Level Knowledge Distillation

Approximate $q(\mathbf{w} | \mathbf{x})$ with mode

$$q(\mathbf{w} | \mathbf{x}) \approx \mathbb{1}\{\arg \max_{\mathbf{w}} q(\mathbf{w} | \mathbf{x})\}$$

Approximate mode with beam search

$$\hat{\mathbf{y}} \approx \arg \max_{\mathbf{w}} q(\mathbf{w} | \mathbf{x})$$

Simple model: train the student model on $\hat{\mathbf{y}}$ with NLL

Sequence-Level Knowledge Distillation

Approximate $q(\mathbf{w} | \mathbf{x})$ with mode

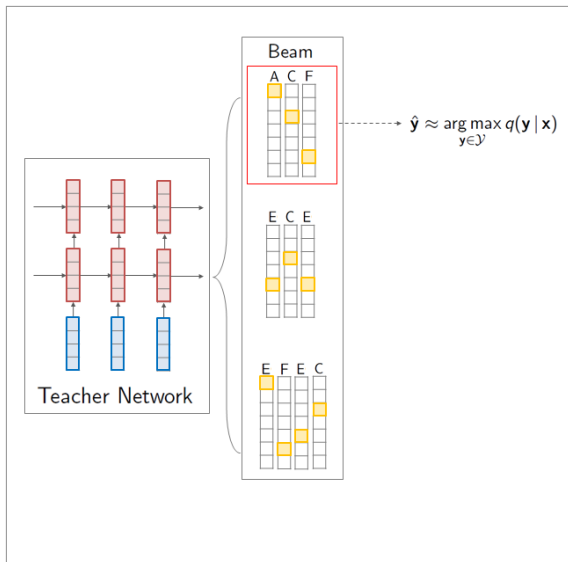
$$q(\mathbf{w} | \mathbf{x}) \approx \mathbb{1}\{\arg \max_{\mathbf{w}} q(\mathbf{w} | \mathbf{x})\}$$

Approximate mode with beam search

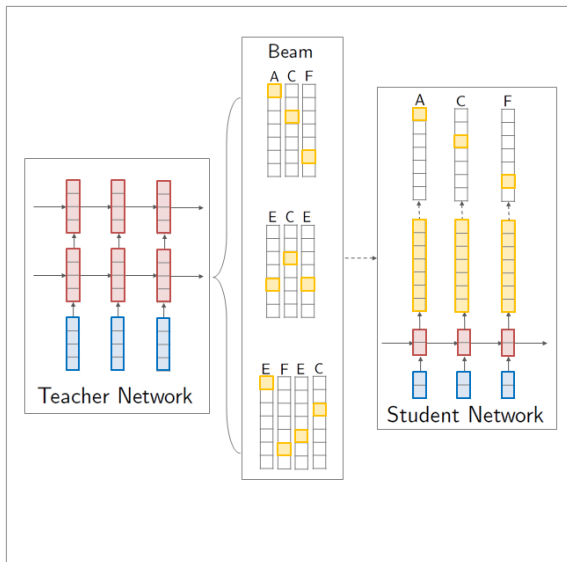
$$\hat{\mathbf{y}} \approx \arg \max_{\mathbf{w}} q(\mathbf{w} | \mathbf{x})$$

Simple model: train the student model on $\hat{\mathbf{y}}$ with NLL

Sequence-Level Knowledge Distillation



Sequence-Level Knowledge Distillation



Sequence-Level Interpolation

Word-level knowledge distillation

$$\mathcal{L} = \alpha \mathcal{L}_{\text{WORD-KD}} + (1 - \alpha) \mathcal{L}_{\text{NLL}}$$

Essentially training the student towards the mixture of teacher/data distributions.

How can we incorporate ground truth data at the sequence-level?

Sequence-Level Interpolation

Naively, could train on both \mathbf{y} (ground truth sequence) and $\hat{\mathbf{y}}$ (beam search output from teacher).

This is non-ideal:

- Doubles size of training set
- \mathbf{y} could be very different from $\hat{\mathbf{y}}$

Consider a *single-sequence* approximation

Sequence-Level Interpolation

Take the sequence that is on the beam but highest similarity function sim (e.g. BLEU) to ground truth

$$\begin{aligned}\tilde{\mathbf{y}} &= \arg \max_{\mathbf{y} \in \mathcal{T}} sim(\mathbf{y}, \mathbf{w})q(\mathbf{w} | \mathbf{x}) \\ &\approx \arg \max_{\mathbf{y} \in \mathcal{T}_K} sim(\mathbf{y}, \mathbf{w})\end{aligned}$$

\mathcal{T}_K : K -best sequences from beam search.

Similar to local updating (Liang et al., 2006)

Train the student model on $\tilde{\mathbf{y}}$ with NLL.

Sequence-Level Interpolation

Take the sequence that is on the beam but highest similarity function sim (e.g. BLEU) to ground truth

$$\begin{aligned}\tilde{\mathbf{y}} &= \arg \max_{\mathbf{y} \in \mathcal{T}} sim(\mathbf{y}, \mathbf{w})q(\mathbf{w} | \mathbf{x}) \\ &\approx \arg \max_{\mathbf{y} \in \mathcal{T}_K} sim(\mathbf{y}, \mathbf{w})\end{aligned}$$

\mathcal{T}_K : K -best sequences from beam search.

Similar to local updating (Liang et al., 2006)

Train the student model on $\tilde{\mathbf{y}}$ with NLL.

Sequence-Level Interpolation

Take the sequence that is on the beam but highest similarity function sim (e.g. BLEU) to ground truth

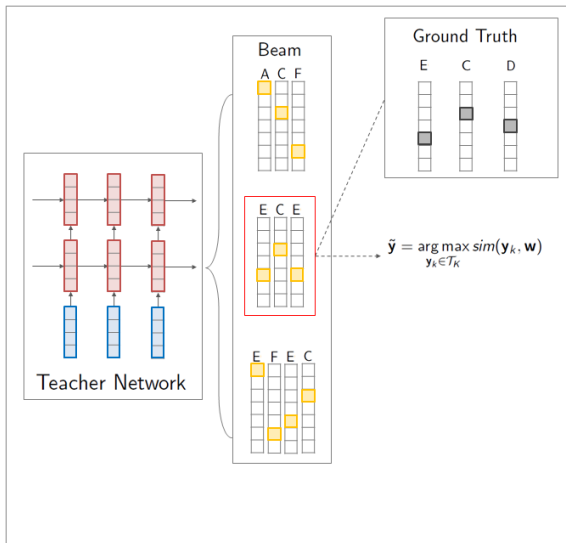
$$\begin{aligned}\tilde{\mathbf{y}} &= \arg \max_{\mathbf{y} \in \mathcal{T}} sim(\mathbf{y}, \mathbf{w})q(\mathbf{w} | \mathbf{x}) \\ &\approx \arg \max_{\mathbf{y} \in \mathcal{T}_K} sim(\mathbf{y}, \mathbf{w})\end{aligned}$$

\mathcal{T}_K : K -best sequences from beam search.

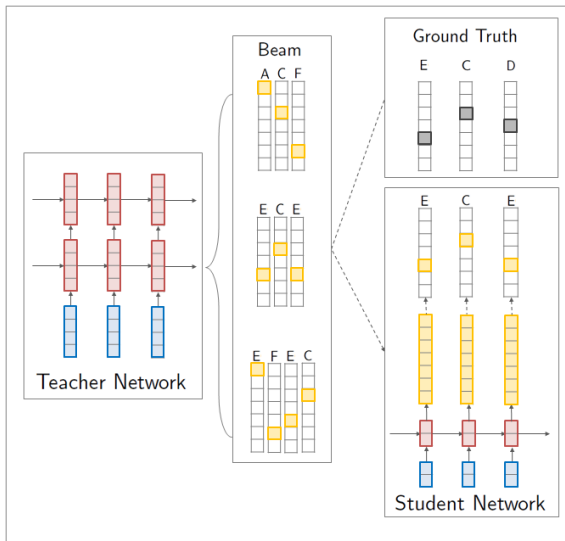
Similar to local updating (Liang et al., 2006)

Train the student model on $\tilde{\mathbf{y}}$ with NLL.

Sequence-Level Interpolation



Sequence-Level Interpolation





Room cancellation is free up to 15 days prior to arrival .

Up to 15 days before arrival are free of charge

Bookings are free of charge 15 days before arrival .

Up to 15 days before arrival , <unk> are free .

Up to 15 days prior to arrival it is free

Up to 15 days before arrival <unk> is free .

Up to 15 days before arrival <unk> are free .

It is free of charge until 15 days before arrival

Up to 15 days before arrival will be free of

Up to 15 days prior to arrival , cancellation charges

Experiments on English → German (WMT 2014)

- Word-KD: Word-level Knowledge Distillation
- Seq-KD: Sequence-level Knowledge Distillation with beam size $K = 5$
- Seq-Inter: Sequence-level Interpolation with beam size $K = 35$.
Fine-tune from pretrained Seq-KD (or baseline) model with smaller learning rate.

Results: English \rightarrow German (WMT 2014)

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(\hat{y})$
<hr/>						
4×1000						
Teacher	17.7	—	19.5	—	6.7	1.3%
<hr/>						
2×500						
Student	14.7	—	17.6	—	8.2	0.9%

Results: English \rightarrow German (WMT 2014)

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(\hat{y})$
4×1000						
Teacher	17.7	–	19.5	–	6.7	1.3%
2×500						
Student	14.7	–	17.6	–	8.2	0.9%
Word-KD	15.4	+0.7	17.7	+0.1	8.0	1.0%

Results: English \rightarrow German (WMT 2014)

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(\hat{y})$
4×1000						
Teacher	17.7	—	19.5	—	6.7	1.3%
2×500						
Student	14.7	—	17.6	—	8.2	0.9%
Word-KD	15.4	+0.7	17.7	+0.1	8.0	1.0%
Seq-KD	18.9	+4.2	19.0	+1.4	22.7	16.9%

Results: English \rightarrow German (WMT 2014)

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(\hat{y})$
4×1000						
Teacher	17.7	—	19.5	—	6.7	1.3%
2×500						
Student	14.7	—	17.6	—	8.2	0.9%
Word-KD	15.4	+0.7	17.7	+0.1	8.0	1.0%
Seq-KD	18.9	+4.2	19.0	+1.4	22.7	16.9%
Seq-Inter	18.9	+4.2	19.3	+1.7	15.8	7.6%

Results: English \rightarrow German (WMT 2014)

Model	$\text{BLEU}_{K=1}$	$\Delta_{K=1}$	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$	PPL	$p(\hat{y})$
4×1000						
Teacher	17.7	—	19.5	—	6.7	1.3%
Seq-Inter	19.6	+1.9	19.8	+0.3	10.4	8.2%
2×500						
Student	14.7	—	17.6	—	8.2	0.9%
Word-KD	15.4	+0.7	17.7	+0.1	8.0	1.0%
Seq-KD	18.9	+4.2	19.0	+1.4	22.7	16.9%
Seq-Inter	18.9	+4.2	19.3	+1.7	15.8	7.6%

Results: English \rightarrow German (WMT 2014)

Model	BLEU _{K=1}	$\Delta_{K=1}$	BLEU _{K=5}	$\Delta_{K=5}$	PPL	$p(\hat{y})$
4×1000						
Teacher	17.7	–	19.5	–	6.7	1.3%
Seq-Inter	19.6	+1.9	19.8	+0.3	10.4	8.2%
2×500						
Student	14.7	–	17.6	–	8.2	0.9%
Word-KD	15.4	+0.7	17.7	+0.1	8.0	1.0%
Seq-KD	18.9	+4.2	19.0	+1.4	22.7	16.9%
Seq-Inter	18.9	+4.2	19.3	+1.7	15.8	7.6%

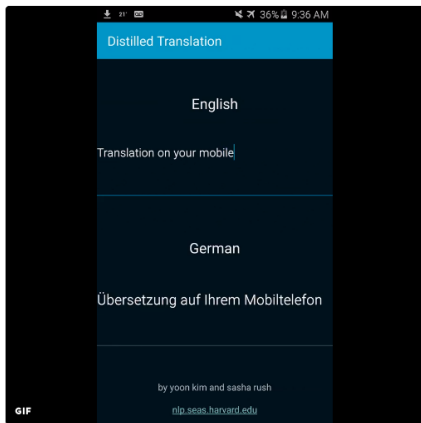
More experiments (different language pairs, combining configurations, different sizes etc.) in paper

An Application

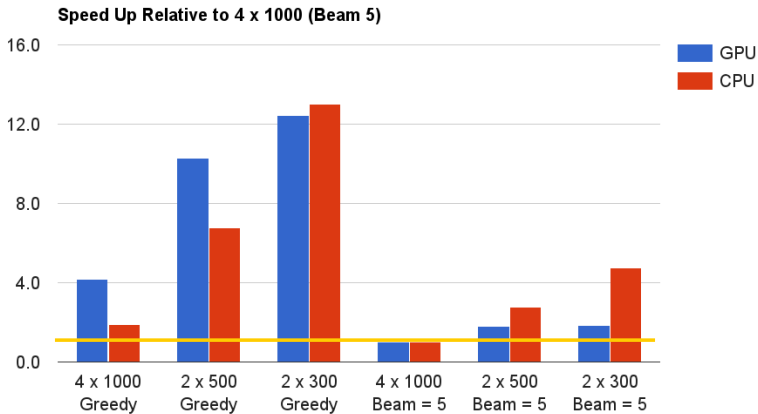


harvardnlp
@harvardnlp

Seq KD (arxiv.org/abs/1606.07947): learn small LSTMs for fast translation. Runs on a phone (nlp.seas.harvard.edu/translation.apk)



Decoding Speed



Combining Knowledge Distillation and Pruning

Number of parameters still large for student models (mostly due to word embedding tables)

- 4×1000 : 221 million
- 2×500 : 84 million
- 2×300 : 49 million

Prune student model: Same methodology as See et al. (2016)

- Prune $x\%$ of weights based on absolute value
- Fine-tune pruned model (crucial!)

Combining Knowledge Distillation and Pruning

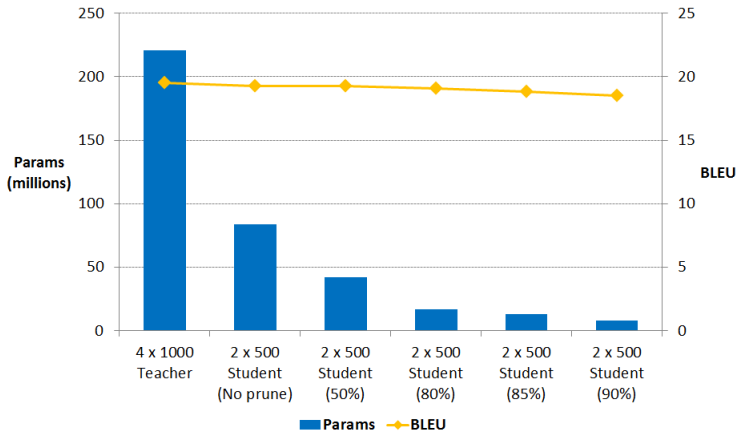
Number of parameters still large for student models (mostly due to word embedding tables)

- 4×1000 : 221 million
- 2×500 : 84 million
- 2×300 : 49 million

Prune student model: Same methodology as See et al. (2016)

- Prune $x\%$ of weights based on absolute value
- Fine-tune pruned model (crucial!)

Combining Knowledge Distillation and Pruning



Conclusion

Introduced sequence-level versions of knowledge distillation to compress NMT models.

Observations:

- Can similarly compress an ensemble into a single model (Kuncoro et al., 2016)
- No beam search \implies we no longer need the softmax at each step: opens up window into approximate inner product methods.

Live deployment: (greedy) student outperforms (beam search) teacher!
(Crego et al., 2016)

Conclusion

Introduced sequence-level versions of knowledge distillation to compress NMT models.

Observations:

- Can similarly compress an ensemble into a single model (Kuncoro et al., 2016)
- No beam search \implies we no longer need the softmax at each step: opens up window into approximate inner product methods.

Live deployment: (greedy) student outperforms (beam search) teacher!
(Crego et al., 2016)

Conclusion

Introduced sequence-level versions of knowledge distillation to compress NMT models.

Observations:

- Can similarly compress an ensemble into a single model (Kuncoro et al., 2016)
- No beam search \implies we no longer need the softmax at each step: opens up window into approximate inner product methods.

Live deployment: (greedy) student outperforms (beam search) teacher!
(Crego et al., 2016)

Thank you

harvardnlp 

<https://github.com/harvardnlp/seq2seq-attn>

Appendix: Decoding Speed

Model Size	GPU	CPU	Android
<i>Beam = 1 (Greedy)</i>			
4×1000	425.5	15.0	–
2×500	1051.3	63.6	8.8
2×300	1267.8	104.3	15.8
<i>Beam = 5</i>			
4×1000	101.9	7.9	–
2×500	181.9	22.1	1.9
2×300	189.1	38.4	3.4

Source words translated per second.

Appendix: Knowledge Distillation and Pruning

Model	Prune %	Params	BLEU	Ratio (Params)
4×1000	0%	221 m	19.5	1×
2×500	0%	84 m	19.3	3×
2×500	50%	42 m	19.3	5×
2×500	80%	17 m	19.1	13×
2×500	85%	13 m	18.8	18×
2×500	90%	8 m	18.5	26×

References I

- Ba, L. J. and Caruana, R. (2014). Do Deep Nets Really Need to be Deep? In *Proceedings of NIPS*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR*.
- Bucila, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model Compression. In *Proceedings of KDD*.
- Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2015). Listen, Attend and Spell. *arXiv:1508.01211*.
- Chen, X., Xu, L., Liu, Z., Sun, M., and Luan, H. (2015). Joint learning of Character and Word Embeddings. In *Proceedings of IJCAI*.

References II

- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of EMNLP*.
- Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-Based Models for Speech Recognition. *arXiv:1506.07503*.
- Crego, J., Kim, J., and Senellart, J. (2016). Systran's pure neural machine translation system. *arXiv preprint arXiv:1602.06023*.
- Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. (2014). Exploiting Linear Structure within Convolutional Neural Networks for Efficient Evaluation. In *Proceedings of NIPS*.
- Gillick, D., Brunk, C., Vinyals, O., and Subramanya, A. (2016). Multilingual Language Processing from Bytes. In *Proceedings of NAACL*.

References III

- Han, S., Mao, H., and Dally, W. J. (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *Proceedings of ICLR*.
- Hermann, K. M., Kcisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching Machines to Read and Comprehend. In *Proceedings of NIPS*.
- Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv:1503.0253*.
- Kuncoro, A., Ballesteros, M., Kong, L., Dyer, C., and Smith, N. A. (2016). Distilling an Ensemble of Greedy Dependency Parsers into One MST Parser. In *Proceedings of EMNLP*.
- LeCun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal Brain Damage. In *Proceedings of NIPS*.

References IV

- Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016). A Diversity-Promoting Objective Function for Neural Conversational Models. In *Proceedings of NAACL 2016*.
- Liang, P., Bouchard-Cote, A., Klein, D., and Taskar, B. (2006). An End-to-End Discriminative Approach to Machine Translation. In *Proceedings of COLING-ACL*.
- Lin, Z., Coubariaux, M., Memisevic, R., and Bengio, Y. (2016). Neural Networks with Few Multiplications. In *Proceedings of ICLR*.
- Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of EMNLP*.
- Rush, A. M., Chopra, S., and Weston, J. (2015). A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of EMNLP*.
- See, A., Luong, M.-T., and Manning, C. D. (2016). Compression of Neural Machine Translation via Pruning. In *Proceedings of CoNLL*.

References V

- Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of AAAI*.
- Srivastava, N., Mansimov, E., and Salakhutdinov, R. (2015). Unsupervised Learning of Video Representations using LSTMs. *Proceedings of ICML*.
- Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS*.
- Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015a). Grammar as a Foreign Language. In *Proceedings of NIPS*.
- Vinyals, O. and Le, Q. (2015). A Neural Conversational Model. In *Proceedings of ICML Deep Learning Workshop*.
- Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015b). Show and Tell: A Neural Image Caption Generator. In *Proceedings of CVPR*.

References VI

- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1606.09.08144*.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of ICML*.
- Zhou, J., Cao, Y., Wang, X., Li, P., and Xu, W. (2016). Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. In *Proceedings of TACL*.